Digital search trees revisited

Philippe Flajolet, Robert Sedgewick

INRIA

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél. (3) 954 90 20

Rapports de Recherche

N° 311

DIGITAL SEARCH TREES REVISITED

Philippe FLAJOLET
Robert SEDGEWICK

Juin 1984

Digital Search Trees Revisited

Philippe FLAJOLET, Robert SEDGEWICK

Résumé :

Plusieurs algorithmes ont été proposés pour construire des arbres de recherche utilisant les propriétés digitales des clefs de recherche. Nous présentons une approche générale à l'analyse en moyenne des propriétés de tels arbres qui conduit à la solution d'un problème ouvert de Knuth. Cet article constitue également un survol de l'analyse des trois principales structures de données fondées sur la recherche digitale : les arbres digitaux dus à Coffman et Eve, les arbres lexicographiques (ou "tries") et les arbres Patricia.

Digital Search Trees Revisited

by

Philippe Flajolet
Robert Sedgewick*

INRIA
Rocquencourt
FRANCE

Abstract:

Several algorithms have been proposed which build search trees using digital properties of the search keys. A general approach to the study of the average case performance of such algorithms is discussed, with particular attention to the analysis of the *digital search tree* structures of Coffman and Eve. Specifically, the method leads to the solution of a problem left open by Knuth, finding the average number of nodes in digital search trees with both sons null.

The paper may be of interest as a survey and tutorial treatment of the analysis of the three primary digital tree search methods: digital search trees, radix search tries, and Patricia tries.

---

* Author's current address: Department of Computer Science, Brown University, Providence, RI. This research was supported in part by NSF Grant MCS-83-08806 and by DARPA under Contract N00014-83-K-0146.

1. Introduction

A fundamental problem in computer science is the so-called *dictionary problem*, where various operations, chiefly *search* and *insert*, are to be performed on a set of records possessing key values. To *insert* a record is to store it away for later retrieval; to *search* is to find a previously stored record with a given key value. The *binary search tree* is an elementary data structure for solving this problem: records are stored in nodes which contain two distinguished values (*left* and *right*) which point to other nodes or could be null. One node, called the *root*, is pointed to by no other nodes, otherwise each node is referenced by exactly one other node. To search for a record with value v, we set x to point to the root and perform the following operations until termination:

- If x is null, then terminate (v not found)
- If $key(x) = v$ then terminate (v found)
- Otherwise, if $v < key(x)$ then set x to $left(x)$;
  if $v > key(x)$ then set x to $right(x)$.

To insert a new record with a new value v we search, then replace the null pointer that caused termination with a pointer to the new record. The analysis of the performance of this method is well known: if records with keys from a random permutation of N elements are successively inserted into an initially empty tree, then the expected number of nodes examined in a successful search in the resulting tree is

$$2\left(1+\frac{1}{N}\right)H_N - 3 = (2\ln 2)\lg N + 2\gamma - 3 + O\left(\frac{\log N}{N}\right).$$

See [9] for details. Throughout this paper we use the notations $H_N = \sum_{1\le k\le N} 1/k = \ln N + \gamma + 1/2N + O(1/N^2)$, where $\gamma = .5772\ldots$ is Euler's constant; $\lg N \equiv \log_2 N$; and $\ln N \equiv \log_e N$. The approximate value of the coefficient of $\lg N$ in the leading term is $1.38630\ldots$. For a perfectly balanced tree, the coefficient would be 1, but an $O(N)$ worst case is possible (for example if the keys are inserted in ascending order). Several methods are available to make the worst case search time close to $\lg N$. One technique is to periodically perform structural modifications on the trees to keep them "well-balanced" (for example, see [16]). In this paper we consider in detail an alternative class of methods.

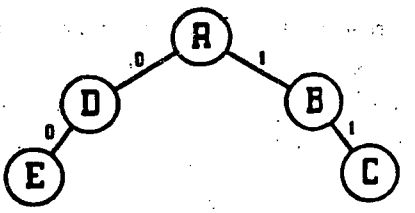
The *digital search tree* [1] is a data structure which leads to much improved worst case performance (and asymptotically optimal average case performance as well) by making use of the digital properties of the keys, if that is appropriate. We simply assume that the keys can be represented as binary numbers so that it makes sense to refer to the b^{th} bit of a key, where the bits are numbered, say, from left to right. Then, to search for a record with value v, we set x to point to the root and b to 1, then perform the following operations until termination:

- If x is null then terminate (v not found)
- If $key(x) = v$ then terminate (v found)
- Otherwise, if the b^{th} bit of v is 0 then set x to $left(x)$;
  if the b^{th} bit of v is 1 then set x to $right(x)$
- Set b to $b+1$.

Insertion is done exactly as with binary search trees (*i. e.*, the null pointer which caused termination is replaced by a pointer to the new record).

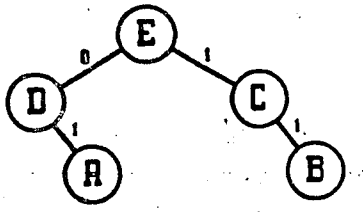
The following is a digital search tree built by inserting the keys at left in the order given:

A: 010
B: 110
C: 111
D: 001
E: 000



Note that the order in which the keys are inserted is relevant. For example, the following tree results from inserting the same keys in reverse order:

E: 000
D: 001
C: 111
B: 110
A: 010



In some implementations, it may be convenient to assume that the keys are all of the same length, as in the examples above. The method also is appropriate for varying length keys, provided that no key is a prefix of another. The number of nodes examined in a digital search tree of N keys is limited by the number of bits in the keys, which is larger than $\lg N$ but is likely to be within a constant factor for many natural situations. The average case performance of this method is also known (in the analysis we assume that the keys are infinitely long): if N records with keys composed of random bit streams are inserted into an initially empty tree, then the average number of nodes examined during a successful search in the resulting digital search tree is

$$\lg N + \frac{\gamma - 1}{\ln 2} + \frac{3}{2} - \alpha + \delta(N) + O\left(\frac{\log N}{N}\right)$$

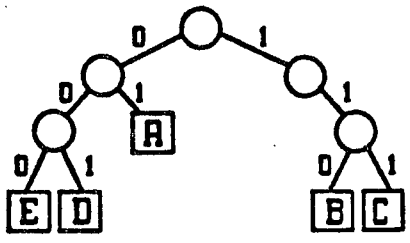
where α is a constant between 1 and 2, and $\delta(N)$ is a small ($|\delta(N)| < 10^{-6}$) oscillatory term ($\delta(2N) = \delta(N)$) which are defined in detail in the next section. This is about 38% less than for binary tree search, but note that the results are not necessarily directly comparable because the input models differ. The above result is due to Konheim and Newman [11]; the refinement including the periodic term is due to Knuth [9, Ex. 6.3–27]. In this paper we give an alternate derivation that generalizes to yield other statistics about the trees, in particular solving a problem left open by Knuth [9, Ex. 6.3–29].

Digital search trees are easily confused with *radix search tries*, a different application of essentially the same idea. A *binary trie* has two types of nodes: *internal nodes*, which consist of left and right pointers only; and *external nodes*, which consist of keys only. To search for a record with value v in a binary trie, we set x to the root and b to 1, then perform the following operations until termination:

- If x is external, then terminate
  (if $key(x) = v$ then v found, otherwise v not found).
- Otherwise, if the b^{th} bit of v is 0 then set x to $left(x)$;
  if the b^{th} bit of v is 1 then set x to $right(x)$.
- Set b to $b + 1$.

Insertion is more complicated in tries than in binary or digital search trees. On termination of an unsuccessful search for a key v to be inserted, we have two keys which belong in the same external node. If the bth bits of those keys differ, we replace that external node by an internal node which points to two external nodes containing the keys; otherwise we have to also include an internal node corresponding to each bit beyond the bth for which the two keys match. The following trie is constructed for the example set of keys above:

A: 010
B: 110
C: 111
D: 001
E: 000



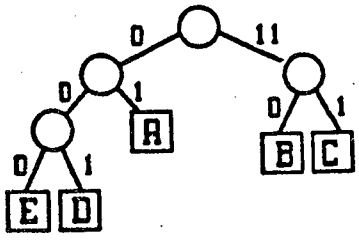
In contrast to digital search trees, the same trie is constructed no matter in what order the keys are inserted. Tries can have more than N internal nodes to store N keys; also handling multiple node types is inconvenient in many programming environments. It is possible to eliminate both of these problems (see below). Most interesting statistics for tries have been fully analyzed; for example, if N records with keys from random bit streams are inserted into an initially empty trie, then the average number of nodes examined during a successful search in the resulting trie is

$$\lg N + \frac{\gamma}{\ln 2} + \frac{1}{2} + \delta(N) + O\left(\frac{1}{N}\right)$$

even though the average number of internal nodes in the trie is about $N/\ln 2 \approx 1.44269\ldots N$.

It is possible to ensure that a trie constructed with N keys has just $(N-1)$ internal nodes by collapsing one-way branches on internal nodes. For our example set we would have:

A: 010
B: 110
C: 111
D: 001
E: 000



Equal bits in keys do not affect the structure of such tries. The programming details of how to accomplish this are not relevant to this paper. We refer to these structures as *Patricia tries* [9] because they are the basis of an algorithm called Patricia which also manages to store the keys in internal nodes and thus avoid the multiple-node-type problem referred to above. Patricia is somewhat more complicated than digital tree searching, but it has applications beyond searching which make it of independent interest. From an analytic standpoint, direct comparisons between Patricia and digital searching are suggested because they both build search keys into (the same class of) binary tree structures, using digital properties of the keys. Knuth has probed many aspects of Patricia in depth: for example, the number of nodes examined in an average successful search is one less than for standard tries.

Many more details on the use and application of these methods may be found in [9] and [16]. In this paper we present new results on the analysis of digital search trees. The above introductory description is intended to motivate this analysis and to provide a context within which we can discuss the relationship of the methods we use to previous analyses of the various algorithms. In the next section, we give an analysis of the average internal path length of a digital search tree which illustrates our basic method and provides an alternate derivation to the one provided by Knuth for this problem. Following that, we use the same general method to find the average number of nodes in a digital search tree with both links null, a somewhat more complicated problem left open by Knuth. In Section 4 we consider M-way branching; Section 5 is a discussion of various generalizations.

2. Path Length

The *internal path length* of a tree is the sum of the number of nodes on the path from the root to each node in the tree. The average number of nodes examined during a successful search in a search tree with N nodes is one plus the internal path length divided by N.

Let A_N be the average internal path length of a digital search tree built from N (sufficiently long) keys comprised of random bits. Then we have the fundamental recurrence relation

$$A_N = N - 1 + \sum_k \frac{1}{2^{N-1}} \binom{N-1}{k} (A_k + A_{N-1-k}), \qquad N \geq 1 \tag{1}$$

with A_0 defined to be 0. This follows from three easily established facts. First, the internal path length of any tree of N nodes is $N-1$ plus the internal path length of the two subtrees of the root. Second, the probability that the left subtree of the root has k nodes (and the right subtree has $N-1-k$ nodes) is $\binom{N-1}{k}/2^{N-1}$, the probability that exactly k of the $N-1$ nodes that are inserted after the first node start with a 0 bit. Third, the subtrees themselves are randomly built according to the same model. Recurrence relations of this type are used to describe the performance of many tree-searching methods. As discussed further below, slight differences in the equations can make the analysis somewhat more intricate.

By symmetry (change k to $N-1-k$ in the second part of the sum), the recurrence (1) is equivalent to

$$A_N = N - 1 + 2\sum_k \frac{1}{2^{N-1}} \binom{N-1}{k} A_k, \qquad N \geq 1$$

with A_0 defined to be 0. This recurrence is transformed into a functional equation on the exponential generating function $A(z) = \sum_{N>0} A_N z^N/N!$ by multiplying both sides by $z^{N-1}/(N-1)!$ and summing for $N \geq 1$:

$$\begin{aligned}
\sum_{N\geq 1} \frac{A_N z^{N-1}}{(N-1)!} &= \sum_{N\geq 2} \frac{z^{N-1}}{(N-2)!} + 2\sum_{N\geq 1} \sum_{0\leq k\leq N-1} \frac{1}{2^{N-1}} \binom{N-1}{k} A_k \frac{z^{N-1}}{(N-1)!} \\
&= ze^z + 2\sum_{k\geq 0} \frac{A_k}{k!} \sum_{N\geq k+1} \left(\frac{z}{2}\right)^{N-1} \frac{1}{(N-k-1)!} \\
&= ze^z + 2\sum_{k\geq 0} \frac{A_k (z/2)^k}{k!} \sum_{N\geq 0} \frac{(z/2)^N}{N!} \\
A'(z) &= ze^z + 2A(z/2)e^{z/2}.
\end{aligned}$$

This difference–differential equation can be transformed into a somewhat more manageable form by introducing the generating function $B(z) = \sum_{N\geq 0} B_N z^N/N!$ with

$$B(z) \equiv e^{-z} A(z).$$

That is, $A(z) = e^z B(z)$ and $A'(z) = e^z B'(z) + e^z B(z)$. In terms of $B(z)$, the above difference-differential equation becomes

$$B'(z) + B(z) = z + 2B(z/2).$$

This corresponds to a simple recurrence on the coefficients

$$B_N + B_{N-1} = \frac{1}{2^{N-2}} B_{N-1},$$

or

$$B_N = -\left(1 - \frac{1}{2^{N-2}}\right) B_{N-1}, \qquad N \geq 3,$$

with $B_2 = 1$, which telescopes to give an explicit formula for B_N:

$$B_N = (-1)^N \prod_{1 \leq j \leq N-2} \left(1 - \frac{1}{2^j}\right).$$

Similar quantities arise in the theory of partitions, and we shall have occasion later to use classical identities from that theory. We have $B_N = (-1)^N Q_{N-2}$, where

$$Q_N \equiv \prod_{1 \leq j \leq N} \left(1 - \frac{1}{2^j}\right).$$

As $N \to \infty$, this approaches the limit $Q_\infty = .288788....$ Now, expanding the formula $A(z) = e^z B(z)$ shows that $A_N = \sum_k \binom{N}{k} B_k$, so (after handling initial conditions) we obtain an explicit formula for A_N:

$$A_N = \sum_{k \geq 2} \binom{N}{k} (-1)^k Q_{k-2}. \tag{2}$$

It remains to evaluate this sum.

At this point, it might be worth noting the relationship between this derivation and the corresponding derivation for binary tries. The fundamental recurrence for tries is

$$A_N^{[T]} = N + \sum_k \frac{1}{2^N} \binom{N}{k} \left(A_k^{[T]} + A_{N-k}^{[T]}\right), \qquad N \geq 2, \tag{3}$$

with $A_0^{[T]}$ and $A_1^{[T]}$ both defined to be 0. This is the number of nodes examined during all successful searches, but it is the average *external* path length of the trie. Note that since no key is stored at the root, the subtrees have a total of N keys. The resulting functional equation on the exponential generating function is not a difference–differential equation but simply a difference equation:

$$A^{[T]}(z) = z(e^z - 1) + 2A^{[T]}(z/2)e^{z/2}.$$

It is still convenient to transform the equation with $A(z) = e^z B(z)$ to get the equation

$$B^{[T]}(z) = z(1 - e^{-z}) + 2B^{[T]}(z/2).$$

This yields directly

$$B_N^{[T]} = \frac{N(-1)^N}{1 - \frac{1}{2^{N-1}}}$$

and

$$A_N^{[T]} = \sum_{k\geq 2} \binom{N}{k} \frac{k(-1)^k}{1-\frac{1}{2^{k-1}}}$$

which is somewhat simpler than (2) and can be handled directly with Mellin transform techniques [2], as described in full detail in [6] and [9, sec. 5.2.2].

The fundamental recurrence for the average external path length of Patricia tries is trickier:

$$A_N^{[P]} = N\left(1-\frac{1}{2^{N-1}}\right) + \sum_k \frac{1}{2^N}\binom{N}{k}\left(A_k^{[P]} + A_{N-k}^{[P]}\right), \qquad N \geq 1, \tag{4}$$

with $A_0^{[P]}$ defined to be 0. The external path length of a Patricia trie is the sum of the external path lengths of the subtries of the root, plus the number of nodes in the subtries (N) *unless* one of the subtries is empty (probability $1/2^{N-1}$). The functional equation on the exponential generating function is similar to that for tries:

$$A^{[P]}(z) = z(e^z - e^{z/2}) + 2A^{[P]}(z/2)e^{z/2}$$

with transformed version

$$B^{[P]}(z) = z(1-e^{-z/2}) + 2B^{[P]}(z/2)$$

which yields directly

$$B_N^{[P]} = \frac{N(-1)^N}{2^{N-1}-1}$$

so that

$$A_N^{[P]} = \sum_{k\geq 2}\binom{N}{k}\frac{k(-1)^k}{2^{k-1}-1} = A_N^{[T]} - N$$

as mentioned above.

The method used above is equivalent to the "binomial transform" method described by Knuth, but it is perhaps more transparent.

For the average internal path length of digital search trees, Knuth uses an approach suggested by Konheim and Newman [11] to transform (2) into a form which has essentially the same asymptotics as the above trie sum. This derivation is somewhat indirect, and does not provide a way to analyze other properties of digital search trees. But Knuth gives an alternate method for evaluating the trie sum, which he attributes to S.O. Rice [9, Ex. 5.2.2-53]. We next show how this method applies directly to (2).

Rice's method is based on a classical formula from the calculus of finite differences [12]:

Lemma 1. *Let C be a closed curve encircling the points $0, 1, \ldots, N$, and let $f(z)$ be a function which is analytic within C. Then*

$$\sum_k \binom{N}{k}(-1)^k f(k) = \frac{1}{2\pi i}\int_C \binom{z}{N}^{-1}(-1)^N f(z).$$

where $\binom{z}{N}$ is the generalized binomial coefficient defined by $\binom{z}{N} \equiv \frac{\Gamma(z+1)}{\Gamma(N+1)\Gamma(z-N)} = \frac{z(z-1)\cdots(z-N)}{N!} = z\prod_{1\leq k\leq N}\left(\frac{z}{k}-1\right)$.

Proof: This is a straightforward application of Cauchy's theorem: the integral is the sum of the residues inside C, and the residue at $z=k$ is $\binom{N}{k}(-1)^k f(k)$ for each k in the range $0 \leq k \leq N$. ∎

This general identity arises in the study of finite differences, since the sum $\sum_k \binom{N}{k}(-1)^k f(k)$ is precisely $\nabla^N f(0)$, where $\nabla f(k) = f(k+1) - f(k)$ (see, for example, [12]).

To use the Lemma for asymptotic analysis, we change C to a large curve around which the integral is small, and take into account residues at poles in the larger enclosed area. This method actually plays a rather fundamental role in the analysis of the class of problems considered here.

Note that the function $\binom{z}{N}^{-1}$ has poles at the integers $0, 1, ..., N$. Thus, using Rice's method with Lemma 1 as stated would involve examining only the singularities of $f(z)$. However, the lemma also clearly holds if the sum is taken over any subset of the integers $0, 1, ..., N$ (and C is taken to enclose just those points): then application of Rice's method might have to take into account the singularities of $\binom{z}{N}^{-1}$ outside C. In particular, in the cases of interest in this paper, the points 0 and 1 are not included in C. In fact, we have to cope with double poles at these points (as well as many singularities for the function $f(z)$).

To apply Rice's method to the evaluation of (2), we need to define an appropriate meromorphic function to extend Q_k, which is defined only on the integers. To this end, we introduce the function

$$Q(x) = \left(1 - \frac{x}{2}\right)\left(1 - \frac{x}{4}\right)\left(1 - \frac{x}{8}\right)\cdots.$$

Note that $Q(1) = Q_\infty$ and that $Q_N = Q(1)/Q(2^{-N})$, so that the analytic expression $f(z) = Q(1)/Q(2^{-z})$, which is defined when z is a positive integer, gives the appropriate extension. Actually, this extension can be derived in a rather mechanical fashion, because Q_N is defined by the recurrence relationship

$$Q_{N+1} = \left(1 - \frac{1}{2^{N+1}}\right) Q_N \qquad N \geq 1,$$

with $Q_0 = 1$, which simplifies to the expression

$$Q_N = \frac{1}{1 - (1/2)^{N+1}} Q_{N+1}.$$

We simply extend this recurrence relation and telescope it:

$$\begin{aligned} f(z) &= \frac{1}{1 - (1/2)^{z+1}} f(z+1) \\ &= \frac{1}{1 - (1/2)^{z+1}} \frac{1}{1 - (1/2)^{z+2}} f(z+2) \\ &\vdots \\ &= \frac{1}{Q(2^{-z})} \lim_{z \to \infty} f(z) \end{aligned}$$

provided the limit exists. But $f(0) = Q_0 = 1$ implies that $\lim_{z\to\infty} f(z) = Q(1)$, so $f(z) = Q(1)/Q(2^{-z})$ is a proper analytic extension. We point out this "mechanical" derivation because we use essentially the same method for a more difficult problem in the next section.

Thus, by Lemma 1 (see also the comments following the lemma), we know that

$$A_N = \frac{1}{2\pi i} \int_C \binom{z}{N}^{-1} (-1)^N \frac{Q(1)}{Q(2^{-z+2})}\, dz \tag{5}$$

where C encloses the points $2, 3, ..., N$. To complete the analysis, we expand C to a larger curve and study the behavior of the integrand at newly enclosed singularities. These residue computations are simplified somewhat because the functions involved have a simple product form; the following lemma, which is elementary, will prove useful for most of the series expansions done in this paper.

Lemma 2. *If $F(z) = \prod_{j\in R}(1 - f_j(z))^{-1}$ for some index set R, then the Taylor series expansion of F at a, if it exists, is given by*

$$F(z) = F(a)\left(1 + \sum_{j\in R}\frac{f_j'(a)}{1 - f_j(a)}(z-a) + O(z-a)^2\right).$$

Proof: Elementary from the use of the "logarithmic derivative": if $G(z) = \prod_{k\in R} g_k(z)$ then $G'(z)/G(z) = \sum_{k\in R} g_k'(z)/g_k(z)$. ∎

For example, at $a = 1$, we have the following expansion for the generalized binomial coefficient:

$$\begin{aligned}(-1)^N\binom{z}{N}^{-1} &= \frac{1}{1-z}z^{-1}\prod_{2\le j\le N}\left(1 - \frac{z}{j}\right)^{-1}\\ &= \frac{1}{1-z}N(1 + (H_{N-1} - 1)(z-1) + O((z-1)^2)\\ &= -\frac{N}{z-1} - N(H_{N-1} - 1) + O(z-1)\end{aligned}$$

Similarly, for the Q function, we have

$$\begin{aligned}Q(1)/Q(2^{-z+1}) &= Q(1)\prod_{j<1}(1 - 2^{-z+j})^{-1}\\ &= 1 - \ln 2\sum_{j<0}\frac{2^{j-1}}{1-2^{j-1}}(z-1) + O(z-1)^2\\ &= 1 - \alpha\ln 2(z-1) + O(z-1)^2.\end{aligned}$$

where $\alpha = 1 + \frac{1}{3} + \frac{1}{7} + \frac{1}{15} + \cdots$. This is a fundamental constant which arises in the analysis of several algorithms, for example Heapsort [9, p. 156] and approximate counting [3].

We are now ready to complete the analysis of the average internal path length of digital search trees using Rice's method for the asymptotic analysis:

Theorem 1 (Konheim–Newman, Knuth). *The average internal path length of a digital search tree built from N records with keys from random bit streams is*

$$(N+1)\lg N + \left(\frac{\gamma - 1}{\ln 2} + \frac{1}{2} - \alpha + \delta(N)\right)N + O(N^{1/2})$$

where $\gamma = .577216\ldots$ is Euler's constant, $\alpha = 1 + \frac{1}{3} + \frac{1}{7} + \frac{1}{15} + \cdots = 1.606695\ldots$, and $\delta(N)$ is a periodic function in $\lg N$, with $|\delta(N)| < 10^{-6}$. The approximate value of the coefficient of the linear term is $-1.7155\ldots$.

Proof: Following the discussion above, the value sought is given by the integral (5). If we change C to a large rectangle R_{XY} with corners at the four points $(\frac{1}{2} \pm iY, X \pm Y)$, then we know by Cauchy's theorem that the integral around R_{XY} (which we shall show to be small) is equal to A_N plus the sum of the residues of

$$\binom{z}{N}^{-1}(-1)^N\frac{Q(1)}{Q(2^{-z+2})}$$

at poles within R_{XY} but not within C.

Rewriting $\binom{z}{N}^{-1}$ as $-\Gamma(N+1)\Gamma(-z)/\Gamma(N+1-z)$, we can make use of standard asymptotic expansions of the Γ function to bound the value of the integral around R_{XY}. We have the approximations

$$\frac{\Gamma(N)}{\Gamma(N+a)} = N^{-a} + O\left(N^{-a-1}\right)$$

and

$$\Gamma(x+iY) = O\left(|Y|^{\frac{x-1}{2}} e^{-\pi|Y|/2}\right)$$

(see, for example, [17, pp. 256–7]). Thus, a bound for our integral along the top and bottom lines of R_{XY} is given by

$$O\left(\int (N+1)^{x+iY} |Y|^{\frac{x-1}{2}} e^{-\pi Y/2} dx\right)$$

This bound is valid only if $Q(2^{-z+2})$ does not get too close to zero; we can insure this by taking Y to be of the form $\frac{\pi}{\ln 2}(2Y'+1)$, with Y' an integer. Thus, the integral is exponentially small in Y and vanishes on the top and bottom of R_{XY} as $Y \to \infty$. A similar argument shows that the integral vanishes on the right of R_{XY} as $x \to \infty$. On the left, we have the bound

$$O\left(\int_{-Y}^{Y} \frac{\Gamma(N+1)}{\Gamma(N+\frac{1}{2}-iy)} dy\right) = O\left(N^{1/2}\right)$$

This proves that A_N plus the sum of the residues in the halfplane to the right of the line $x = 1/2$ is $O(N^{1/2})$. We now proceed to calculate these residues.

The integrand has poles at $z = j \pm \frac{2\pi i k}{\ln 2}$ for $j = 1, 0, -1, -2, \ldots$, and all $k \geq 0$, since at these points $2^{-z+j} = 1$ which causes one of the factors of $Q(2^{-z+2})$ to vanish. The poles at 0 and 1 are double poles because $\binom{z}{N}^{-1}$ is also singular at 0 and 1. Of these, only the poles for $j = 1$ are within the region of interest; thus we have to compute residues at the double pole 1 and at the points $1 \pm \frac{2\pi i k}{\ln 2}$ for $k \neq 0$.

At $z = 1$, we use the series expansions derived from Lemma 2 above:

$$\begin{aligned}
\binom{z}{N}^{-1} (-1)^N \frac{Q(1)}{Q(2^{-z+2})} &= \binom{z}{N}^{-1} (-1)^N \frac{1}{1-2^{-z+1}} \frac{Q(1)}{Q(2^{-z+1})} \\
&= \left(-\frac{N}{z-1} - N(H_{N-1}-1)(z-1) + O(z-1)^2\right) \times \\
&\quad \left(\frac{1}{(z-1)\ln 2} + \frac{1}{2} + O(z-1)\right) \times \\
&\quad \left(1 - \alpha \ln 2 (z-1) + O(z-1)^2\right).
\end{aligned}$$

The residue at $z = 1$ (the coefficient of $1/(z-1)$ in this product) is

$$\frac{N}{\ln 2}(H_{N-1}-1) + N\left(\alpha - \frac{1}{2}\right) = -N \ln N - N\left(\frac{\gamma - 1}{\ln 2} - \alpha + \frac{1}{2}\right) + O(1).$$

The poles at $1 \pm \frac{2\pi i k}{\ln 2}$ add a small contribution to the linear term: the total residue at these points is

$$\frac{1}{\ln 2} \sum_{k \neq 1} \binom{1 + \frac{2\pi i k}{\ln 2}}{N}^{-1} (-1)^N.$$

As above, we can write $\binom{z}{N}^{-1}(-1)^N$ as $-\Gamma(N+1)\Gamma(-z)/\Gamma(N+1-z)$ and use Lemma 2 to develop an asymptotic expansion. We have

$$\begin{aligned}\binom{1+\frac{2\pi ik}{\ln 2}}{N}^{-1}(-1)^N &= -\frac{\Gamma(N+1)\Gamma\left(-1-\frac{2\pi ik}{\ln 2}\right)}{\Gamma\left(N-\frac{2\pi ik}{\ln 2}\right)} \\ &= -N\Gamma\left(-1-\frac{2\pi ik}{\ln 2}\right)\frac{\Gamma(N)}{\Gamma\left(N-\frac{2\pi ik}{\ln 2}\right)} \\ &= -N\Gamma\left(-1-\frac{2\pi ik}{\ln 2}\right)N^{\frac{2\pi ik}{\ln 2}}+O(1).\end{aligned}$$

Thus, the sum of the residues at the points $1\pm\frac{2\pi ik}{\ln 2}$ is

$$-N\delta(N)+O(1)$$

where

$$\delta(N)=\frac{1}{\ln 2}\sum_{k\neq 0}\Gamma\left(-1-\frac{2\pi ik}{\ln 2}\right)e^{2\pi ik\lg N}.$$

This and related functions arise in the study of many algorithms, for example: evaluating arithmetic expressions [5], parallel addition [10], extendible hashing [13], approximate counting [3], and Batcher's merging networks [15]. The properties of $\delta(N)$ cited in Theorem 1 are discussed in Knuth [9, p. 134].

Thus, subtracting the sum of the residues at $z=1$ and at $z=1\pm\frac{2\pi ik}{\ln 2}$ for $k\neq 0$ from the estimate of the value of the contour integral around R_{XY} in the limit, we have shown that

$$A_N=N\ln N+N\left(\frac{\gamma-1}{\ln 2}-\alpha+\frac{1}{2}+\delta(N)\right)+O(N^{1/2})$$

as desired. The same method of analysis can be used to expand A_N to any desired asymptotic accuracy, by using a contour R_{XY} which includes more poles. The double pole at $z=0$ and the poles at $z=\pm\frac{2\pi ik}{\ln 2}$ for $k\neq 0$ contribute a constant term like the coefficient of the linear term, and the poles at $z=j\pm\frac{2\pi ik}{\ln 2}$ for $j=-1,-2,\ldots$ contribute more complicated (but very small) oscillatory terms. ∎

Our interest in this derivation is that it illustrates a general method of evaluating sums of the form $\sum_k\binom{N}{k}(-1)^k f(k)$ even when $f(k)$ is a relatively complicated function. (As mentioned above, the proof of Theorem 1 is quite specific to Q_{k-2}.) Essentially, the asymptotic analysis is reduced to a singularity analysis of a meromorphic function satisfying the same recurrence as $f(k)$. Next, we show how this method applies for a function satisfying an inhomogeneous recurrence. This problem arises naturally in the study of other properties of digital search trees.

3. External Internal Nodes

A property of trees of some interest is the number of internal nodes which have both links null. An alternate storage representation could be used for such nodes. The question left open by Knuth (see [9, Ex. 6.3-29]) is to determine exactly how much storage can be saved. This is of more practical importance when M-ary trees (not binary) are considered (see the next section). In this paper we are interested in the problem chiefly because it illustrates the power of Rice's method, as contrasted with standard Mellin techniques, which seem difficult to apply directly to this problem. (Another application of Rice's method may be found in [7].)

In a fully balanced binary tree of N nodes the number of nodes with both links null is $\lceil\frac{N}{2}\rceil$; in a completely unbalanced tree the number is 1. It is a simple exercise to show that the average number of such

nodes for a random binary search tree is $(N+1)/3$. We expect digital search trees to be somewhat more balanced than binary search trees; thus the result should be that somewhere between one-third and one-half of the nodes have both links null. It is mildly surprising that the answer is somewhat closer to the former than the latter.

Theorem 2. *The average number of nodes with both links null in a digital search tree built from N records with keys from random bit streams is*

$$N\left(\beta+1-\frac{1}{Q_\infty}\left(\frac{1}{\ln 2}+\alpha^2-\alpha\right)+\delta^*(N)\right)+O(N^{1/2})$$

where the constants involved have the values $\alpha = 1+\frac{1}{3}+\frac{1}{7}+\frac{1}{15}+\cdots = 1.606695\ldots$, $Q_\infty = \frac{1}{2}\cdot\frac{3}{4}\cdot\frac{7}{8}\cdots = .288788\ldots$, *and* $\beta = \frac{1\cdot 2^2}{1}\left[\frac{1}{1}\right]+\frac{2\cdot 2^3}{1\cdot 3}\left[\frac{1}{1}+\frac{1}{3}\right]+\frac{3\cdot 2^4}{1\cdot 3\cdot 7}\left[\frac{1}{1}+\frac{1}{3}+\frac{1}{7}\right]+\cdots = 7.74313\ldots$. *The function $\delta^*(N)$ is a periodic function in $\lg N$, with $|\delta^*(N)| < 10^{-6}$. The approximate value of the coefficient of the leading term is* $0.372046812\ldots$.

Proof: As before, we use a simple transform with generating functions to derive an explicit sum, then use Rice's method to evaluate the sum.

If C_N is the quantity sought, then we have the recurrence

$$C_N = \sum_k \frac{1}{2^{N-1}}\binom{N-1}{k}(C_k + C_{N-1-k}), \qquad N \geq 2, \tag{6}$$

with $C_1 = 1$ and $C_0 = 0$. This follows from the same argument as for (1), with the additional observation that the number of nodes with both links null in a tree is exactly the sum of the numbers of such nodes in the two subtrees of the root, unless the tree has just one node.

In terms of the exponential generating function $C(z) = \sum_{N\geq 0} C_N z^N/N!$, this leads to the equation

$$C'(z) = 1 + 2C(z/2)e^{z/2}$$

which becomes, in terms of the transform generating function $D(z) = \sum_{N\geq 0} D_N z^N/N!$, with $D(z)$ defined to be $e^{-z}C(z)$:

$$D'(z) + D(z) = e^{-z} + 2D(z/2)$$

This gives a recurrence on the coefficients as before:

$$D_N + D_{N-1} = (-1)^{N-1} + \frac{1}{2^{N-2}}D_{N-1}$$

$$D_N = (-1)^{N-1} - \left(1 - \frac{1}{2^{N-2}}\right), \qquad N \geq 2,$$

with $D_0 = 0$ and $D_1 = 1$. But this recurrence is inhomogeneous, so the telescoped solution is somewhat more complicated than before:

$$D_N = (-1)^{N-1}\sum_{1\leq i\leq N-1}\prod_{i\leq j\leq N-2}\left(1-\frac{1}{2^j}\right)$$

Rewriting this in terms of

$$R_N \equiv Q_N\left(1+\frac{1}{Q_1}+\cdots+\frac{1}{Q_N}\right)$$

and transforming by $C(z) = e^z D(z)$ we have the following explicit sum for the desired quantity:

$$C_N = N - \sum_{k\geq 2} \binom{N}{k} (-1)^k R_{k-2} \tag{7}$$

This sum is more difficult to evaluate than (2) because R_N is more complicated than Q_N.

We might begin by mimicking the "mechanical" derivation of $Q(z)$, turning the recurrence defining R_k around to define a meromorphic function satisfying the same recurrence. We have

$$R_{N+1} = 1 + \left(1 - \frac{1}{2^{N+1}}\right) R_N$$

or, solving for R_N and substituting $q = 1/2$:

$$R_N = -\frac{1}{1-q^{N+1}} + \frac{1}{1-q^{N+1}} R_{N+1}. \tag{8}$$

(From this point on, we will use q for $1/2$. Not only is this a notational convenience, but we'll see in the next section that this is the only change necessary to solve the same problem for M-ary digital search trees.) Using this recurrence to extend to a function on the complex plane would give

$$\begin{aligned} R(z) &= \frac{1}{1-q^{N+1}} + \frac{1}{1-q^{N+1}} R(z+1) \\ &= \sum_{j\geq 0} \frac{1}{(1-q^{z+1})(1-q^{z+2})\cdots(1-q^{z+1+j})}. \end{aligned}$$

Unfortunately, this sum does not converge when z is a positive integer, so it certainly doesn't extend R_N. The reason is clear: R_N itself is not bounded as N increases, so extending a recurrence to increasing N is doomed to failure. Fortunately, it is not difficult to avoid this problem: by studying the asymptotic performance of R_N we can find a closely related function which can be extended by the above technique.

This asymptotic development is elementary from Lemma 2 because the generating function for R_N/Q_N is closely related to a classical identity in the theory of partitions:

Lemma 3 (Euler).

$$\sum_{n\geq 0} \frac{u^n}{(1-q)(1-q^2)\cdots(1-q^n)} = \frac{1}{(1-u)(1-qu)(1-q^2u)\cdots}.$$

Proof: The coefficient of $u^n q^m$ on both sides is the number of ways to write n as the sum of m nonnegative integers. (See Hardy and Wright [8] for related identities and many more details.) ∎

In the notation that we've been using, this identity says that $\sum_{N\geq 0} u^N/Q_N = 1/(1-u)Q(u)$. This gives a convenient way to rewrite the generating function $R(u) = \sum_{N\geq 0}(R_N/Q_N)u^N$ in product form:

$$R(u) = \frac{1}{1-u} \sum_{N\geq 0} \frac{u^N}{Q_N} = \frac{1}{(1-u)^2 Q(u)}$$

Now we can expand $Q(u)^{-1}$ by Lemma 2:

$$\frac{1}{Q(u)} = \frac{1}{Q_\infty} + \frac{\alpha}{Q_\infty}(1-u)$$

Thus $R(u) = 1/(Q_\infty(1-u)^2) + \alpha/(Q_\infty(1-u))$ plus a function which is meromorphic for $|u| < 2$, which implies that

$$R_N = Q_N\left(\frac{1}{Q_\infty(N+1)} - \frac{\alpha}{Q_\infty}\right) + O(2^{-N})$$

Since $Q_N/Q_\infty = 1 + \mathcal{O}(2^{-N})$ this simplifies to

$$R_N = N + 2 - \alpha + O(2^{-N})$$

Now, the function $R_N^* = R_N - (N + 1 - \alpha)$ not only satisfies a simple recurrence but also converges very quickly, so we can apply the recurrence for increasing N to extend the function. From (8) we have

$$R_N^* = \frac{(N+1+\alpha)q^{N+1}}{1-q^{N+1}} + \frac{1}{1-q^{N+1}}R_{N+1}^* \tag{9}$$

which is extended by the meromorphic function

$$\begin{aligned} R(z) &= \frac{(z+1+\alpha)q^{z+1}}{1-q^{z+1}} + \frac{1}{1-q^{z+1}}R(z+1) \\ &= \frac{(z+1+\alpha)q^{z+1}}{1-q^{z+1}} + \frac{(z+2+\alpha)q^{z+2}}{(1-q^{z+1})(1-q^{z+2})} + \frac{1}{1-q^{z+2}}R(z+2) \\ &= \sum_{j\geq 0}\frac{(z+1+j+\alpha)q^{z+1+j}}{(1-q^{z+1})(1-q^{z+2})\cdots(1-q^{z+1+j})}. \end{aligned} \tag{10}$$

This function is meromorphic except for simple poles at the points $z = j \pm \frac{2\pi ik}{\ln 2}$ for $j \leq -2$ and all $k \geq 0$, and for $j = -1$ and $k > 0$. Note carefully that $R(-1)$ exists (why?).

Substituting in (7), we have

$$C_N = N - \sum_{k\geq 2}\binom{N}{k}(-1)^k(R_{k-2}^* + k - 1 - \alpha).$$

The extra terms are easily evaluated from the elementary identities $\sum_k \binom{N}{k}(-1)^k = \sum_k \binom{N}{k}k(-1)^k = 0$ to give the simplified result

$$C_N = (N+1)(\alpha+1) - \sum_{k\geq 2}\binom{N}{k}(-1)^k R_{k-2}^*. \tag{11}$$

Now, by Lemma 1 we know that

$$C_N - (N+1)(\alpha+1) = -\frac{1}{2\pi i}\int_C \binom{z}{N}^{-1}(-1)^N R(z-2)dz. \tag{12}$$

The same argument as for Theorem 1 shows that the right-hand side of this equation is equal to the sum of the residues of the integrand at singularities to the right of the line $x = 1/2$, to within $O(N^{1/2})$. In this case, the poles at $1 \pm \frac{2\pi ik}{\ln 2}$ are all single poles. The main term is given by $N \lim_{z\to 1} R(z-2)$; the poles for $k \neq 0$ add a small oscillatory term.

The method of calculating $R(-1)$ is to express $R(z)$ in terms of a generating function which generalizes the function of Lemma 3, then to expand that function and exploit certain properties of its derivatives. Specifically, we define

$$F(u,v) = \sum_{j\geq 1}\frac{q^j u^j}{(1-qv)\cdots(1-q^j v)}$$

This is the generating function for restricted partitions (the coefficient of $u^n v^m q^k$ is the number of ways to partition n into at most m parts not exceeding k). Note that, by Lemma 3,

$$F(u,1)+1=\frac{1}{(1-qu)(1-q^2u)(1-q^3u)\cdots}$$

so that $F(1,1)=Q_\infty^{-1}-1$. Also, from Lemma 2 we have

$$F_1'(u,1)=\frac{1}{(1-qu)(1-q^2u)(1-q^3u)\quad\cdots}\left(\frac{q}{1-qu}+\frac{q^2}{1-q^2u}+\frac{q^3}{1-q^3u}+\cdots\right)$$

so that $F_1'(1,1)=\alpha/Q_\infty$. Furthermore, we have

$$F(1,q^{z+1})=\sum_{j\geq 1}\frac{q^j}{(1-q^{z+2})\cdots(1-q^{z+1+j})}$$

$$F_1'(1,q^{z+1})=\sum_{j\geq 1}\frac{jq^j}{(1-q^{z+2})\cdots(1-q^{z+1+j})}$$

which gives, from (10), the following expansion:

$$R(z)=\frac{q^{z+1}}{1-q^{z+1}}\left((z+1-\alpha)(F(1,q^{z+1})+1)+F_1'(1,q^{z+1})\right).$$

From this formulation, a Taylor expansion around $z=-1$ is straightforward:

$$\frac{q^{z+1}}{1-q^{z+1}}=-\frac{1}{(z+1)\ln q}-\frac{1}{2}+\frac{(z+1)}{2}\ln q+O(z+1)^2$$

$$F(1,q^{z+1})=F(1,1)+(z+1)\ln qF_2'(1,1)$$

and

$$F_1'(1,q^{z+1})=F_1'(1,1)+(z+1)\ln qF_{12}''(1,1)$$

so that

$$R(z)=-\frac{F(1,1)+1}{\ln q}+\alpha F_2'(1,1)-F_{12}''(1,1)+O(z+1).$$

Note carefully the cancellation of $-\alpha(F(1,1)+1)+F_1'(1,1)$; this is also implied by the fact that $R(z)$ exists at $z=-1$. Thus, to complete the calculation of the main term, we need only calculate $F_2'(1,1)$ and $F_{12}''(1,1)$. These are constants which can easily be calculated from the series representations

$$F_2'(1,1)=\sum_{j\geq 1}\frac{q^j}{(1-q)(1-q^2)\cdots(1-q^j)}\left(\frac{q}{1-q}+\frac{q^2}{1-q^2}+\cdots+\frac{q^j}{1-q^j}\right)$$

$$F_{12}''(1,1)=\sum_{j\geq 1}\frac{jq^j}{(1-q)(1-q^2)\cdots(1-q^j)}\left(\frac{q}{1-q}+\frac{q^2}{1-q^2}+\cdots+\frac{q^j}{1-q^j}\right). \tag{14}$$

Actually, we can relate $F_2'(1,1)$ to α and Q_∞, for the function $F(u,v)$ has a symmetry property which seems remarkable from an analytic standpoint (though it is more intuitive from the combinatorial interpretation). We have

$$\begin{aligned}F(u,v)-vF(qu,v)&=\sum_{k\geq 1}\frac{q^ku^k(1-q^kv)}{(1-qv)\cdots(1-q^kv)}\\&=qu(1+F(u,v)).\end{aligned}$$

This recurrence can be telescoped as follows:

$$\begin{aligned} F(u,v) &= \frac{qu}{1-qu} + \frac{v}{1-qu}F(qu,v) \\ &= \frac{qu}{1-qu} + \frac{vq^2u}{(1-qu)(1-q^2u)} + \frac{v^2}{(1-qu)(1-q^2u)}F(q^2u,v) \\ &= uF(v,u)/v \end{aligned}$$

or $vF(u,v) = uF(v,u)$. (See [14] for some related identities and techniques.)

Differentiating both sides of this symmetric identity with respect to u, we find that $F_1'(1,1) = F_2'(1,1) + F(1,1)$, so $F_2'(1,1) = (\alpha - 1)Q_\infty^{-1} + 1$. Note that differentiating again with respect to v produces a trivial identity: there does not seem to be an easy way to express $F_{12}''(1,1)$ in terms of α and Q_∞, so we denote that constant (defined in (14)) simply by β. Collecting terms, we have shown that the residue of the integrand in (12) at $z = 1$ is

$$\beta + 1 - \frac{1}{Q_\infty}\left(\frac{-1}{\ln q} + \alpha^2 - \alpha\right)$$

It remains to calculate the residues of the integrand in (12) at the other singularities.

This calculation is straightforward: the residue of $(1-q^{z+1})^{-1}$ at $z = -1 \pm \frac{2\pi ik}{\ln q}$ is $-1/\ln q$, and the other terms in $R(z)$ contribute a factor of $2\pi ik/Q_\infty \ln q$. The factor from $\binom{z}{N}^{-1}(-1)^N$ is expanded exactly as for Theorem 1; thus we have the oscillatory term

$$\delta^*(N) = \frac{1}{Q_\infty \ln q}\sum_{k\neq 0}\frac{2\pi ik}{\ln q}\Gamma\left(-1-\frac{2\pi ik}{\ln q}\right)e^{2\pi ik \lg N}.$$

This completes the calculation of the coefficient of the linear term. ∎

For purposes of comparison, it is of some interest to compute the average number of internal nodes with both sons external in Patricia tries. This is a relatively straightforward derivation similar to the path length calculations given above, so we only sketch it here. We start with the recurrence

$$C_N^{[P]} = \sum_k \frac{1}{2^N}\binom{N}{k}\left(C_k^{[P]} + C_{N-k}^{[P]}\right), \qquad N \geq 3 \tag{15}$$

with $C_0^{[P]} = C_1^{[P]} = 0$ and $C_2^{[P]} = 1$. This corresponds to the functional equation

$$C^{[P]}(z) = (z/2)^2 + 2C^{[P]}(z/2)e^{z/2}$$

which transforms to

$$D^{[P]}(z) = (z/2)^2 e^{-z} + 2D^{[P]}(z/2)$$

and eventually gives the sum

$$C_N^{[P]} = \frac{1}{4}\sum_{2\leq k\leq N}\binom{N}{k}\frac{k(k-1)(-1)^k}{1-\frac{1}{2^{k-1}}}.$$

Knuth [9, Ex. 6.3-19] gives specific evaluations of such sums. The eventual result is that the proportion of nodes in Patricia tries with both sons external is $1/(4\ln 2) = .3606...$ plus a small oscillating term. Thus, according to this measure, digital search trees are (slightly) more balanced than Patricia tries.

4. Multiway branching

A natural generalization of the digital tree search (and trie search) algorithm studied above is to allow M-way branching (not just left and right), each node containing M links to other nodes. If $M = 2^m$ and the keys are in binary, this is conveniently implemented as follows. To search for a record with value v, set x to point to the root and b to 1, and perform the following operations until termination:

- If x is null then terminate (v not found).
- If $key(x) = v$ then terminate (v found).
- Otherwise, if the $m(b-1)+1$ through mb^{th} bits of v represent k then set x to the kth link of x.
- Set b to $b+1$.

Here each node is assumed to have M links, numbered from 0 to $M-1$. A similar implementation of tries (with 0 keys and M links per node) is straightforward. (The generalization of binary search trees is quite different: M-ary search trees have $M-1$ keys and M links per node.)

It turns out that the analysis given above survives largely intact for the M-ary case. For example, to find the average number of nodes in a M-ary digital search tree with all links null, we begin with the fundamental recurrence:

$$C_N^{[M]} = \sum_{k_1+k_2+\cdots+k_M=N} \frac{1}{M^{N-1}} \binom{N-1}{k_1, k_2, \cdots, k_M} \left(C_{k_1}^{[M]} + C_{k_2}^{[M]} + \cdots + C_{k_M}^{[M]}\right) \qquad N \geq 2, \tag{16}$$

with $C_1^{[M]} = 1$ and $C_0^{[M]} = 0$. This is proven in the same manner as (1) and (6). First, the number of nodes with all links null in a tree is exactly the number of such nodes in all the subtrees of the root, unless the tree has just one node. Second, the probability that the first subtree has k_1 nodes, the second has k_2 nodes, etc. with $k_1 + k_2 + \ldots + k_M = N-1$ is exactly

$$\frac{1}{M^{N-1}} \binom{N-1}{k_1, k_2, \cdots, k_M}$$

Third, all the subtrees are randomly built according to the same model.

By symmetry, (16) is equivalent to

$$C_N^{[M]} = M \sum_{k_1+k_2+\cdots+k_M=N} \frac{1}{M^{N-1}} \binom{N-1}{k_1, k_2, \cdots, k_M} C_k^{[M]}, \qquad N \geq 2,$$

with $C_1^{[M]} = 1$ and $C_0^{[M]} = 0$. Now, by manipulations generalizing those leading to (2), we define the exponential generating function $C^{[M]}(z) = \sum_{N\geq 0} C_N^{[M]} z^N/N!$ and derive the following difference-differential equation:

$$\begin{aligned} \frac{d}{dz} C^{[M]}(z) &= 1 + MC^{[M]}(z/M) \left(e^{z/M}\right)^{M-1} \\ &= 1 + MC^{[M]}(z/M) e^{(1-\frac{1}{M})z} \end{aligned} \tag{17}$$

For $M = 2$, this is exactly the equation derived from (6); moreover, none of the manipulations used for solving (6) depend in an essential way on the value of that constant. In fact, we defined $q = 1/2$ for notational convenience in that derivation: if we take $q = 1/M$ in the solution, we get the solution for M-ary digital trees.

Corollary. *The average number of nodes with all links null in an M-ary search tree (for $M \geq 2$) built from N records with keys from random bit streams is*

$$N\left(\beta^{[M]} + 1 - \frac{1}{Q^{[M]}_\infty}\left(\frac{1}{\ln M} + \alpha^{[M]}(\alpha^{[M]} - 1)\right) + \delta^{[M]}(N)\right) + O(N^{1/2})$$

where the constants involved are given by

$$Q^{[M]}_\infty = \prod_{k\geq 1}\left(1 - \frac{1}{M^k}\right)$$

$$\alpha^{[M]} = \sum_{k\geq 1}\frac{1}{M^k - 1}$$

$$\beta^{[M]} = \sum_{k\geq 1}\frac{kM^{k+1}}{(M-1)(M^2-1)\cdots(M^k-1)}\sum_{1\leq j\leq k}\frac{1}{M^j - 1}$$

and the oscillatory term is

$$\delta^{[M]}(N) = \frac{1}{Q_\infty \ln M}\sum\frac{2\pi i k}{\ln M}\Gamma\left(-1 + \frac{2\pi i k}{\ln M}\right)e^{2\pi i k \lg N}$$

Proof: Immediate from the discussion above. ∎

The table below gives the approximate values of $C^{[M]}_N$ and the various constants for small values of M.

M	$Q^{[M]}_\infty$	$\alpha^{[M]}$	$\beta^{[M]}$	$C^{[M]}_N/N$
2	.28879	1.60670	7.74313	.37205
3	.56013	.68215	.71399	.47602
4	.68854	.42110	.22414	.53054
8	.85941	.16097	.02748	.62506
16	.93359	.07085	.00510	.68928

Note carefully that, in a perfectly balanced M-ary tree, about $(M-1)/M$ of the nodes have all links null. Measured against this standard, the constants in the last column of the table show that digital search trees are about 70%-75% balanced for small M. That is, the ratio between the constants given and $(M-1)/M$ is between .70 and .75. Of course, as $M \to \infty$, this ratio approaches 1.

5. General framework

The methods that we have used in the previous sections can be applied to study many other properties of digital search trees. If $X(T)$ and $x(T)$ are parameters of trees satisfying

$$X(T) = \sum_{\substack{\text{subtrees } T_j \\ \text{of the root of } T}} X(T_j) + x(T) \tag{18}$$

then the exponential generating functions for the expectations X_N and x_N for an M-ary digital search tree built from N records with keys from random bit streams satisfy

$$X'(z) = MX(z/M)e^{(1-\frac{1}{M})z} + x(z).$$

This is derived in exactly the same manner as (17). Now, in terms of the transform generating functions $Y(z) = e^{-z}X(z)$ and $y(z) = e^{-z}x(z)$ this becomes

$$Y'(z) + Y(z) = MY(z/M) + y'(z) + y(z). \tag{19}$$

This leads to a nonlinear recurrence like (8) satisfied by Y_N, with the solution sought given by $X_N = \sum_k \binom{N}{k} Y_k$. If the quantity $(-1)^k Y_k$ is sufficiently well-behaved, we can study its asymptotics and find a function Y_k^* which:

(i) is simply related to Y_k so that $\sum_k \binom{N}{k} (Y_k - (-1)^k(Y_k^*))$ is easily evaluated,
(ii) satisfies a recurrence of the form $Y_{N+1}^* = (1 - g(M,N))\, Y_N^* + f(M,N)$
(iii) goes to zero quickly as $N \to \infty$.

Depending on the nature of $g(M,N)$, $f(M,N)$ and the speed of convergence, conditions (ii) and (iii) may allow the recurrence to be turned around to extend Y_N^* to the complex plane and so allow the desired expectation to be computed by evaluating the sum $\sum_k \binom{N}{k} (Y_k - (-1)^k(Y_k^*))$ as detailed in the previous sections.

For example, this method could be used to find the distribution of occupancy of nodes in M-ary digital search trees, and many other problems.

The same type of generalization applies to the study of tries (and Patricia tries), and the simpler nature of the recurrences follows through the generalization. For example, the exponential generating functions for the expectations X_N and x_N of parameters of trees satisfying (18) for a random trie built from N records from random bit streams is

$$X(z) = MX(z/M)e^{z/M} + x(z) \tag{20}$$

which is considerably easier to deal with.

These methods allow quite full analysis of the types of trees considered, and they clearly expose the fundamental differences and similarities among the analyses.

A final note: the reader who is still awake may have noticed that the "transforms" that are essential to these computations are not at all arbitrary functions. Indeed, if $x(z) = \sum_{N \geq 0} x_N z^N/N!$ is the exponential generating function for the expectation of a parameter X, then $Y(z) = e^{-z}x(z)$ is the expectation of X if the number of keys is Poisson with parameter z.

For digital search trees, tries, and Patricia tries, this function satisfies a simple functional equation like (20) which makes it amenable to direct solution by Mellin transform techniques. (See [6] for details of the application of this method to the analysis of tries: essentially the Mellin transform is trivially computed by taking the transform of both sides of the functional equation, and then a singularity analysis is done for the reverse transform. Another example of this technique may be found in [4].) The relationships among the Bernoulli and Poisson models and Mellin transform and Rice's method of asymptotic analysis are a fruitful area for further study. More details will be reported in a future paper.

Acknowledgement

The authors would like to express their graditude to Janet Incerpi, who is still awake.

References

[1] E. G. Coffman, Jr. and J. Eve, "File structures using hashing functions," *Communications of the ACM* 13, 7(1970), 427–436.

[2] E. Davies, *Integral Transforms and Their Applications*, Springer-Verlag, 1978.

[3] P. Flajolet, "Approximate counting: a detailed analysis," *BIT*, to appear.

[4] P. Flajolet and C. Puech, "Partial match retrieval of multildimensional data," INRIA Research Report, 1983.

[5] P. Flajolet, J. C. Raoult, and J. Vuillemin, "On the average number of registers required for evaluating arithmetic expression," *Theoretical Computer Science* 9, (1979), 99–125.

[6] P. Flajolet, M. Regnier, and R. Sedgewick, "Mellin Transform Techniques for the Analysis of Algorithms," in preparation.

[7] P. Flajolet and R. Sedgewick, "The asymptotic evaluation of some alternating sums involving binomial coefficients," INRIA Research Report, 1983.

[8] G. Hardy and E. Wright, *An Introduction to the Theory of Numbers*, Clarendon Press, Oxford, (1960).

[9] D. E. Knuth, *The Art of Computer Programming. Volume 3: Sorting and Searching*, Addison-Wesley, Reading, Mass. (1973).

[10] D. E. Knuth, "The average time for carry propagation," 238–242.

[11] A. G. Konheim and D. J. Newman, "A note on growing binary trees," *Discrete Mathematics* 4 (1973), 57–63.

[12] N. E. Nörlund, *Vorlesungen über Differenzenrechnung*, Chelsea Publishing Company, New York (1954).

[13] M. Regnier, "Evaluation des performances du hachage dynamique," Thèse de 3ème cycle, Université de Paris-Sud, 1983.

[14] G. Pólya and G. Szegö, *Problems and Theorems in Analysis I*, Springer-Verlag, Berlin, 1976.

[15] R. Sedgewick, "Data Movement in Odd-Even Merging," *SIAM J. Computing* 7, 3, August 1978.

[16] R. Sedgewick, *Algorithms*, Addison-Wesley, Reading, Mass., 1983.

[17] E. Whittaker and G. Watson, *A course of modern analysis*, Cambridge University Press, 1927.